Docket No. SJO920010066US1
Firm No. 0037.0017

METHOD, SYSTEM, AND ARTICLE OF MANUFACTURE FOR IMPLEMENTING SECURITY FEATURES AT A PORTAL SERVER

BACKGROUND OF THE INVENTION

1. Field of the Invention

[0001] The present invention relates to a method, system, program, and article of manufacture for implementing security features at a portal server.

2. Description of the Related Art

[0002] A portal site is a World Wide Web site or service that offers a broad array of resources and services, such as e-mail, forums, search engines, and on-line shopping malls. A portal server functions as a Web server that hosts the portal site. Prior art portal sites usually categorize content and provide a hyperlink for each category. The hyperlinks may lead to other Internet Web sites outside the portal server. Users access the portal server via a Web browser and click on a hyperlink to read content. Examples of such portal servers are those run by Yahoo!**, Microsoft** Network, and America Online**.

[0003] Some portal servers provide access to a plurality of software applications, where the software applications are stored in servers that are external to the portal server. Such software applications are called backend applications, and the servers in which the backend applications are stored are called backend systems. A user directs a Web browser to connect to the portal server, and subsequently accesses the backend applications via the portal server. The portal servers provide a single point of interaction to the backend applications personalized to the user's needs and responsibilities. A single unified interface on a portal server typically provides the single point of interaction to a user.

[0004] Portal servers can transform the manner in which users access, manage, and share essential data and applications. Portal servers may organize business applications, syndicated content, e-mail messages, and any other relevant information into a workspace

that can be customized to a user's specifications. An example of such a portal server is the Netegrity** Interaction Server.

[0005] When a portal server provides access to backend applications users do not have to store bookmarks at a Web browser for each of the individual backend applications. For example, corporate users may use a Web browser and access corporate-wide applications, such as Web-based electronic mail, instant messaging system, corporate accounting information etc., via a corporate portal server.

[0006] Users may have to authenticate with a portal server by typing in a username and a password, using a smartcard or via other means before the users can access the backend applications. Some prior art portal servers also use the authentication information of the user to display personalized information tailored for the user. Hence, prior art portal servers provide a rudimentary security mechanism for authentication with the portal server before allowing access to the backend applications.

[0007] However, even in prior art portal servers that require authentication from a user, a backend application may require additional authentications before users can access the backend application. In addition, there are security issues beyond authentication. For example, even within the same backend application different users may have different types of privileges. For example, some users may be able to update corporate accounting information whereas other users may be able to read but not update the corporate accounting information.

[0008] There are existing single sign-on products, such as the Novell** Single Sign-on or the Tivoli** Global Sign-on that enable client applications to authenticate with a plurality of servers via a single login. However, such single sign-on applications generally do not enforce a high level of security beyond authentication and furthermore are directory-based software solutions. In addition, such directory-based single sign-on products do not typically function on a portal server.

[0009] Hence, there is a need in the art to provide a system, method and article of manufacture for a portal server that securely allows access to a plurality of backend applications stored on backend servers.

SUMMARY OF THE PREFERRED EMBODIMENTS

[0010] Provided are a method, system and article of manufacture for implementing security features at a portal server. The portal server receives a first request from a client. In response to receiving the first request, the portal server authenticates the client. The portal server consults a database to determine access privileges of the authenticated client for interactions with a plurality of applications, wherein the applications are located at backend servers. Then the portal server generates code containing selectable interactions with the applications, wherein any authentication for the selectable interactions can be performed within the portal server. Subsequently, the portal server sends the code to the client.

[0011] In one implementation, in response to sending the code to the client, the portal server receives a second request from the client, wherein the second request contains a selection of at least one of the selectable interactions. The portal server determines from the selection a set of backend servers to process the request. The portal server forwards the request to the set of backend servers. The portal server receives results corresponding to the request from applications executing on the backend servers and sends the results to the client. In one implementation, along with the results, further selectable operations are sent to the client.

[0012] Additional implementations describe a method, system and article of manufacture for securely distributing a backend application from a backend system. The backend system creates data structures corresponding to interactions with the backend application. The backend system associates privileges for each of the data structures, wherein the privileges can be checked at a portal application separately hosted from the backend application. The backend application receives a request from the portal application for reading the data structures and sends the data structures to the portal application. In one implementation, the backend application receives a request for an interaction with the backend application from the portal application. The backend

application processes the request without checking for the privileges and sends the results of processing the request to the portal server.

[0013] Additional implementations describe a method, system, and article of manufacture for accessing a group of applications at a client computer. The client computer authenticates with a portal server. Subsequently, the client computer receives from the portal server a list of interactions that can be performed with the applications, wherein the applications are stored at backend servers that are different from the portal server. The client computer selects an interaction and receives results based on the selection of the interaction without authenticating with the backend servers. In one implementation, the client receives a set of further interactions selectable by the client.

[0014] The implementations enable a client to access a plurality of backend application via a single portal server. The portal server enforces security on behalf of the plurality of backend applications.

BRIEF DESCRIPTION OF THE DRAWINGS

[0015] Referring now to the drawings in which like reference numbers represent corresponding parts throughout:

FIG. 1 illustrates a block diagram of a computing environment including certain implementations of the invention;

FIG. 2 illustrates the data structures associated with a generic objects database in a portal server in accordance with certain described implementations of the invention;

FIG. 3 illustrates logic to populate a generic objects database in a portal server in accordance with certain described implementations of the invention;

FIG. 4 illustrates logic to process the method for accessing a backend application from a browser in accordance with certain described implementations of the invention; and

FIG 5 illustrates a graphical user interface on a browser in accordance with certain described implementations of the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0016] In the following description, reference is made to the accompanying drawings which form a part hereof and which illustrate several implementations. It is understood that other implementations may be utilized and structural and operational changes may be made without departing from the scope of the present implementations

[0017] FIG. 1 illustrates a block diagram of a computing environment including certain implementations of the invention. A portal server 100 contains a portal application 102 and connects to two networks 104 and 106. The portal server 100 may be any computational device such as a personal computer, a workstation, a server-class computer, a mainframe, a laptop, hand-held, palm-top or telephony device. Network 104 and 106 may be a local area network, an Intranet, the Internet or any other type of network. In one implementation network 104 is a local area network and network 106 is the Internet.

[0018] Portal server 100 is located within a demilitarized zone (DMZ) 108. The DMZ 108 allows the portal server 100 to host Internet services but at the same time prevents unauthorized access to the network 104 via Internet connections to the portal server 100. Computational devices that connect to network 106 cannot connect to computational devices that connect to network 104 except via the portal server 100. The DMZ 108 insulates network 104 and 106 from each other and thereby provides some network security. The DMZ 108 is created by insulating the portal server 100 via firewalls, proxy servers etc. from networks 104, 106 in a manner known in the art.

[0019] The portal application 102 is a Web based application. Clients 110 and 112 can connect to the portal application 102 on the portal server 100 through the network 106 via the hypertext transfer protocol (HTTP) from Web browsers 114, 116. For example, Web browser 114 may send a HTTP request for the portal application 102 from client 110 to portal server 100 across network 106. In response to the HTTP request from the client 110, the portal application 102 sends a Web page to the client 110. The Web browser 114 on the client 110 displays the Web page. The portal application may be implemented in any programming language such as Java**, C++ etc. The Web pages sent by the portal

server 100 to the clients 110 and 112 may include code in Active server pages**, Java server pages, Hypertext Markup languages (HTML), Extensible Markup Language (XML) etc. The Web browsers 114, 116 render the code on the screen of the clients 110, 112.

[0020] Backend systems 118, 120, 122 connect to portal server 100 via the network 104. Each of the backend systems 118, 120, 122 contains one or more backend application [1...*w*] 124, 126, 128, 130. In FIG. 1, backend system 118, contains one backend application 124; backend system 120 contains two backend applications 126, 128; and backend system 122 contains one backend application 130. The backend systems 118, 120, 122 may be any computational device such as a personal computer, a workstation, a server-class computer, a mainframe, a laptop, hand-held, palm-top or telephony device. The backend applications 124, 126, 128, 130 may be any server-based software application such as Web-based electronic mail, an Instant messenger server, a server-based spreadsheet, a database server etc.

[0021] The portal application 102 provides a single point of access to the [1...*w*] backend applications 124, 126, 128, 130. Clients 110, 112 access the [1...*w*] backend applications 124, 126, 128, 130 by accessing the portal application 102.

[0022] FIG. 2 illustrates the data structures associated with a generic objects database in a portal server in accordance with certain implementations of the invention. The backend application *p* 128 [where *p* varies from 1...*n*] has *n* secure data objects [1...*n*] 200, with the i^{th} secure data object denoted by reference numeral 200i. Each of the *n* secure data objects 200 may have different security attributes. For example, a secure data object *i* 200i may be readable and writable by one user but inaccessible to another user. Each of the secure data objects 200 may correspond to different interactions with the backend application *p* 128, different resources of the backend application *p* 128, different operations with the backend application *p* 128, or different types of functions within the backend application *p* 128.

[0023] Associated with the portal application 102 is a portal generic objects database 203. For every backend application *p* 128, the portal generic objects database 203

contains a set of generic objects and associated security attributes for each potential user. In FIG. 2, for backend application *p* 128, the portal generic objects database 203 contains *n* elements, such as *n* generic objects [1...*n*] 204, where the generic object *i* 204i corresponds to secure data object *i* 200i. The generic object *i* 204i stores at least those parts of the secure data object *i* 200i that are necessary for accessing and manipulating the secure data object *i* 200i. The portal generic objects database 203, stores for each generic object *i* 204i the security attributes for *r* users [1...*r*] 206, with the i^{th} user denoted by reference numeral 206i. For example, in FIG. 2 for backend application *p* 128, secure data object *i* 200i, represented by generic object *i* 204i is not accessible (shown in table entry marked with reference numeral 208) to user *i* 206i.

[0024] FIG. 2 has shown only certain security attributes corresponding to the secure data objects 200 in the portal generic objects database 203. Furthermore, the portal generic objects database 203 has been represented as a table. Alternative implementations where the portal generic objects database 203 is represented in forms other than a table, such as in a relational database format, in extensible markup language (XML), a class etc., and where the security attributes are different from those shown in FIG. 2 are possible.

[0025] FIG. 3 illustrates logic to populate a portal generic objects database 203 in a portal server 100 in accordance with certain implementations of the invention. If there are *w* backend applications [1...*w*] 124, 126, 128, 130, each of the backend applications [1...*w*] 124, 126, 128, 130 initially create (in blocks 300a, 300b, 300c) generic objects 204 corresponding to the secure data objects 200 of each of the backend applications [1...*w*] 124, 126, 128, 130. The secure data objects 200 contain the privileges and other security attributes related to each user of the corresponding secure data objects. In certain implementation, the secure data objects can also be modeled as business objects, where the business objects encapsulate security policies and practices of an enterprise.

[0026] Subsequent to the creation of the secure data objects 200 by the backend applications, the portal application 102 initializes (at block 302) the portal generic objects database 203. At block 304, the portal application 102 assigns the variable *p* to one. The

portal application 102 reads (at block 306) the secure data objects 200 of backend application *p* 128. The portal application 102 adds (at block 308) generic objects 204 to the portal generic objects database 203 corresponding to the secure data objects 200 of backend application *p* 128. The portal application 102 and the backend application *p* 128 must have a prior arrangement such that the format of the generic objects 204 created by the backend application *p* 128 are interpretable by the portal application 102. Different backend applications [1...*w*] 124, 126, 128, 130 may have different a priori arrangements with the portal application 102. In one implementation the secure data objects 200 may be in a standardized data representation format such as an extensible markup language (XML) format. Other data structure representations, such relational databases, classes may be also used to implement the secure data objects 200 in a manner known in the art.

[0027] Control passes to block 310, where the portal application 102 determines if generic objects 204 have been created for all backend applications [1...*w*] 124, 126, 128, 130. If so, at block 312 the process comes to a stop. Otherwise, control proceeds to block 314 where the portal application 102 increments *p*, and repeats the logic of blocks 306 to 310 for the next backend application, i.e. backend application p. In an alternative implementation instead of the portal application 102 updating the generic objects database, the backend applications [1...*w*] 124, 126, 128, 130 update the generic objects database.

[0028] Subsequent to executing the logic block 312, the backend applications [1...*w*] 124, 126, 128, 131 may change, such as when a new version of a backend application [1...*w*] 124, 126, 128, 130 is installed on a backend system 118, 120, 122. In such case, control proceeds to block 316 where the portal application 102 incrementally updates the generic objects database 203 to reflect the change in the backend application [1...*w*] 124, 126, 128, 130.

[0029] FIG. 4 illustrates logic implemented in the Web browser 114 on the client 110, the portal application 102 and the backend application 130 to access the backend application 130 from the Web browser 114 in accordance with certain implementations of the invention. The Web browser 114 uses the portal application 102 to access the backend

application 130. In other implementations, the Web browser 114 may also access additional backend applications via the portal application 102. In further implementations, the portal application 102 may also allow other Web browsers on other clients to access the backend application 130, and other backend applications. Thus a plurality of clients may access a plurality of backend applications by the portal application 130.

[0030] At block 402, the Web browser 114 located on the client 110 sends a request to the portal application 102. The portal application 102 receives (at block 404) the request. The portal application 102 requests (at block 406) authentication information from the client 110. The client 110 receives (at block 408) the request for authentication. Control proceeds to block 410, where a user at the client 110 enters the authentication information and the client 110 sends the authentication information to the portal application 102. The authentication information may include a username and password combination, data on a smartcard etc. The portal application 102 receives (at block 412) the authentication information entered by the user from the client 110 and may establish a secure session with the client 110 in a manner known in the art, thereby avoiding repeated entry of authentication information by the user. In alternative implementations, a computer program may perform the functions of the user. At block 414, the portal application 102 refers to the generic objects database 203 to determine access privileges of authenticated user.

[0031] Control proceeds to block 416, where the portal application 102 generates a page to the client 110 containing entries corresponding to the backend applications [1...*w*] 124, 126, 128, 130 that the authenticated user can access, based on the access privileges of the authenticated user. The page may also include interactions, such as the associated operations that the authenticated user can perform and the resources the authenticated user can access based on the access privileges of the authenticated user. The portal application sends (at block 417) the page to the client 110. Additional details of the page are described in FIG. 5.

[0032] FIG 5 illustrates a graphical user interface on a Web browser in accordance with certain implementations of the invention. A Web browser 114 at client 110 receives code corresponding to a Web page 502 from portal application 102 and displays the page 502 at the client 110. The Web page 502 may contain a header text 504 stating "User *i*, welcome to the portal application" followed optionally by results of earlier operations 505 by User *i*, followed by instructions 506 requesting User *i* to "select operation(s) and/or resource(s)."

[0033] The Web browser 114 also displays on page 502 a list of backend applications 128, 130 along with operations 510, 512 that can be performed on each backend application 128, 130 and resources 513, 514 that can be selected on each backend application 128, 130. Resources 513, 514 could include any entity related to the backend applications 128, 130 such as objects, data structures, files etc. For example, in backend application *p* 128, User *i* may perform the operations 510, i.e. read and write on data *1* and read on data *2, 3,* and *4.* User *i* may also select resource *1* 513. The page 502 corresponds to the page sent by the portal application 102 in blocks 417, 432 of the logic illustrated in FIG. 4. The graphical user interface illustrated in FIG. 5 may be displayed in a variety of other ways, including different selection mechanisms such as hyperlinks, radio buttons, check boxes etc. In addition in some implementations, the user *i* may be able to select a multiple number of operations, resources, or any other interaction with a backend application from the graphical user interface.

[0034] Returning back to the description of FIG. 4, the entries corresponding to the resources and operations the user may access on the page sent by the portal application 102 may include hyperlinks, items in a drop down list, selectable radio buttons associated with the entries or any other user interface controls. The page may include active code such as in Java, in addition to HTML code. For example, the portal server may send a page containing an entry, where the entry indicates that for a particular corporate financial backend program the authenticated user is entitled to read but not update the financial information contained within the financial backend program.

[0035] Control proceeds to block 418, where the authenticated user at client 110 receives the page from the portal application 102. The authenticated user selects one of the entries in the page the authenticated user has received and requests (at block 420) an operation or a resource within the corresponding backend application [1...*w*] 124, 126, 128, 130. The portal application 102 receives (at block 422) the request for access to the operation or the resource from the authenticated user.

[0036] At block 424, the portal application 102 determines that the operation or resource is part of application *p* 128 and forwards the request to the backend application *p* 128. The backend application *p* 128 receives (at block 426) the request for the operation or resource within the backend application *p*128. The backend application *p* 128 generates the result of the operation or the selection of the resource and sends (at block 428) the result to the portal application 102. The operations may include any operation that can be performed by a software application, such as a database query, a file transfer, an instant message interaction etc. Similarly, the resources can include any resource related to the backend application *p* 128. The backend application *p* 128 may secure the resource from another backend application or may perform the operations via interactions with another backend application.

[0037] The portal application 102 receives (at block 430) the result from the backend application *p* 128 and sends (at block 432) the result, including entries for further operations or resources to the authenticated user. The authenticated user receives (at block 434) the result and the entries for further operations and resources. The authenticated user may continue to select a further operation or resource on Web browser 114 in which case control proceeds to block 420 and the logic of blocks 422 to 434 is repeated. If at block 434 the authenticated user after receiving the result and the entries for further operations and resources does not select an operation or resource, the process may terminate at block 434. The process may also terminate if the user does not select an operation or resource within a predetermined amount of time via a session timeout in a manner known in the art.

[0038] The implementations enable a portal server to ensure security on behalf of backend applications. Unlike prior art access mechanisms for a backend application wherein a user on a client may have to a priori know the details pertaining to the backend application before accessing the application, the portal server allows the user to access backend applications residing on backend servers without any such a priori knowledge. The client has to perform authentications only with the portal server and does not have to repeatedly authenticate with each backend server. Hence, with the described implementations, the clients can access the backend applications faster, a request from a client can be mapped by the portal server to multiple backend applications, and incremental changes to backend applications can be accommodated by incremental changes to the portal application.

Additional Implementation Details

[0039] The described portal applications, backend applications and client applications may be implemented as a method, apparatus or article of manufacture using standard programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof. The term "article of manufacture" as used herein refers to code or logic implemented in hardware logic (e.g., an integrated circuit chip, Field Programmable Gate Array (FPGA), Application Specific Integrated Circuit (ASIC), etc.) or a computer readable medium (e.g., magnetic storage medium, hard disk drives, floppy disks, tape, etc.), optical storage (CD-ROMs, optical disks, etc.), volatile and non-volatile memory devices (e.g., EEPROMs, ROMs, PROMs, RAMs, DRAMs, SRAMs, firmware, programmable logic, etc.). Code in the computer readable medium is accessed and executed by a processor. The code in which implementations are made may further be accessible through a transmission media or from a file server over a network. In such cases, the article of manufacture in which the code is implemented may comprise a transmission media, such as a network transmission line, wireless transmission media, signals propagating through space, radio waves, infrared signals, etc. Of course, those skilled in the art will recognize that many modifications may be made to this

configuration without departing from the scope of the implementations, and that the article of manufacture may comprise any information bearing medium known in the art.

[0040] The implementations can be adapted for any type of networked data delivery mechanisms, including client-server and peer-to-peer data delivery mechanisms. Also, in alternative implementations, some of the backend applications may be stored in the portal server. Although the implementations have been described with reference to a Web browser at a client, other display mechanisms besides a Web browser, e.g. a client window such as an X-window, can substitute or augment the Web browser.

[0041] The resources the backend applications make available to the users may comprise multimedia content, objects, files, attributes of objects, program elements, database objects, table entries, data structures etc. or any other types of resources known in the art. The operations may include any operations such as database queries, program execution, functions etc. or any other types of operations known in the art.

[0042] The preferred logic of FIGs. 3 and 4 described specific operations occurring in a particular order. Further, the steps may be performed in parallel as well as sequentially. In alternative embodiments, certain of the logic operations may be performed in a different order, modified or removed and still implement preferred embodiments of the present invention. Morever, steps may be added to the above described logic and still conform to the preferred embodiments.

[0043] Therefore, the foregoing description of the implementations has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. It is intended that the scope of the invention be limited not by this detailed description, but rather by the claims appended hereto. The above specification, examples and data provide a complete description of the manufacture

and use of the composition of the invention. Since many embodiments of the invention can be made without departing from the spirit and scope of the invention, the invention resides in the claims hereinafter appended.

** Yahoo! is a trademark of Yahoo!, Inc.; Microsoft and Active Server Pages are trademarks of Microsoft Corporation; America Online is a trademark of America Online, Inc.; Netegrity is a trademark of Netegrity, Inc.; Novell is a trademark of Novell, Inc.; Tivoli is a trademark of Tivoli Systems, Inc.; Java and Java server pages are trademarks of Sun Microsystems, Inc.